CRYPTANALYSIS OF CRYPTOGRAPHIC PRIMITIVES AND RELATED TOPICS

CRYPTANALYSIS OF CRYPTOGRAPHIC PRIMITIVES AND RELATED TOPICS

Seyed Mehdi
Mohammad Hassanzadeh

DISSERTATION FOR
THE DEGREE OF PHILOSOPHIAE DOCTOR

THE SELMER CENTER
DEPARTMENT OF INFORMATICS
UNIVERSITY OF BERGEN
NORWAY

JUNE 2011

تقدیم به پدر، مادر و همسر عزیزم

ABSTRACT

This thesis has focused on the cryptanalysis of cryptographic primitives especially stream ciphers which is an important topic in cryptography. Additionally, the security of network coding is discussed and improved with a new scheme.

First, a new statistical test, called *Quadratic Box-Test*, is presented. It can be used to evaluate the randomness quality of the pseudorandom sequences which can be the output of a cryptographic primitive. Moreover, it can be used as a distinguisher to attack stream ciphers, block ciphers and hash functions.

In the second part of the thesis, some stream ciphers are analyzed and some successful attacks are presented. A modified algebraic attack is used against some clock controlled stream ciphers. In order to have successful attacks, the modified algebraic attack is accompanied by some new ideas. Moreover, the security of clock controlled stream ciphers based on its jumping system is investigated and discussed which resulted in some recommendations to design a clock controlled stream cipher. Finally, a differential distinguishing attack based on a fault attack is presented in this thesis to attack the Shannon stream cipher.

The last part of this thesis focuses on the security of network coding which promises increased efficiency for future networks. For secure network coding, a new attack model is studied and the secrecy capacity is improved by a concatenated secret sharing scheme.

ACKNOWLEDGEMENTS

During my study in Norway, I have received an invaluable amount of support and encouragement from many individuals. It is impossible to name all who have made valuable contributions along the way, but I wish to use this opportunity to thank a few.

First and foremost, I would like to express my immense appreciation to my supervisor, Tor Helleseth, for his knowledge, inspiration, enthusiasm and excellent supervision. It is for sure that this thesis would not have been possible without you.

Secondly, I would like to extend my thanks to my co-supervisor, Igor Semaev, has been a constant source of enthusiasm and ideas, and his guidance has been invaluable. Thank you for giving me time and attention whenever requested, and always taking my concerns seriously.

Special thanks to Mohammad Ravanbakhsh and Mohammad Reza Sohizadeh for all the great discussions. And also, I would like to thank all my other collaborators at the Informatics Department for providing an excellent atmosphere for research.

Finally, my sincere acknowledgment is to my family, for being an irreplaceable source of support and guidance in every step of my life and my studies.

CONTENTS

LIST OF FIGURES

1 MOTIVATION

The history of cryptography begins thousands of years ago. The earliest known use of cryptography is found in non-standard hieroglyphs carved into monuments from Egypt's Old Kingdom (ca 4500+ years ago). There are also many manuscripts in cryptography and cryptanalysis written during the golden age of Islam. For example, two philosophers during that age were *Ahmad al-Qalqashandi* and *Al-Kindi* that contributed and published manuscripts in cryptography and cryptanalysis.

The *Subh al-a'sha* is a 14-volume encyclopedia which includes a section on cryptology and written by Ahmad al-Qalqashandi (1355-1418 AH). This work included a list of ciphers consisting of both *substitution* and *transposition*, and for the first time, a cipher with *multiple substitutions* for each plaintext letter. Also traced to *Ibn al-Duraihim* is an exposition and worked example of cryptanalysis, including the use of tables of letter frequencies and sets of letters which do not appear consecutively.

Around AD 800, Al-Kindi [1], an Arab mathematician, invented the frequency analysis technique for breaking mono-alphabetic substitution ciphers which was probably religiously motivated from textual analysis of the **Qur'an**, the Islamic holy book. He wrote a book on cryptography entitled *Risalah fi Istikhraj al-Mu'amma* (Manuscript for the Deciphering of Cryptographic Messages), in which he gave the first descriptions on frequency analysis. It is the first documented occurrence of systematic cryptanalysis techniques and also it includes some poly-alphabetic ciphers, cipher classification and most importantly describes the use of several statistical techniques for cryptanalysis [57]. It also covered methods of encipherments, cryptanalysis of certain encipherments, and statistical analysis of letters and letter combinations in Arabic [1, 2]. Al-Kindi's work was the most fundamental cryptanalytic advance until WWII.

During WWII, Claude Shannon introduced his theory which was the starting point of *Modern Cryptography* that evolved considerably as a science. He published his theory in 1949 entitled "Communication Theory of Secrecy Systems" which established a solid theoretical basis for cryptography and also for cryptanalysis based on "information and communication theory".

Modern cryptography can be divided in two classes; *symmetric* and *asymmetric* cryptography. In symmetric cryptography, sender and re-

ceiver use the same key and it is more often used for transmitting larger quantities of data than asymmetric cryptography where the sender and receiver use different keys. Asymmetric cryptography is often used to establish a secret key between the sender and receiver. In symmetric cryptography, the security depends on the security of the common key, e.g. *cryptographic primitives* (stream cipher, block cipher, hash function, ...).

Cryptographic primitives play the most important role in secure system design as the basic blocks. Frequent use of cryptographic primitives in military, governments and industries is the best evidence for their importance. Due to their importance, there are several competitions to standardize block ciphers, stream ciphers and hash functions. On January 2, 1997, NIST (National Institute of Standards and Technology of the United States [50]) announced a competition for block cipher to choose a successor to DES to be known as AES and it was completed in October 2000. *eSTREAM/ECRYPT* was a project to identify a new stream cipher as a standard and organized by the EU ECRYPT network [22]. The *NESSIE* project [51] was its ancestor but all submitted candidates were rejected. The call for stream ciphers was first issued in November 2004 and completed in April 2008.

After my graduation in cryptography as a Master student, I started my activity in cryptanalysis of stream ciphers in the eSTREAM project. After some successful attacks on several candidates [31, 32, 42–44], I started my PhD studies in 2007 and continued my cryptanalysis experience. This resulted in some successful attacks on several stream ciphers and these results are presented in my thesis.

Recently, there is another competition but this time for the primitive "hash function" organized by NIST. It was formally announced on November 2, 2007 and the proclamation of a winner and publication of the new standard are scheduled to take place in 2012. This competition was contemporaneous with my PhD study and I found a chance to contribute to this topic and presented a new statistical test which can be used as a distinguisher for hash functions and other primitives.

2 BACKGROUND

There are many concepts in cryptography, but a brief overview of some related topics to this thesis are given in this section. Readers who are already familiar with these concepts can skip the corresponding parts.

2.1 RANDOMNESS

One of the most important topics in cryptography is randomness and randomness measurement. The security of most cryptographic systems depends upon a random sequence. For example, the secret key in block ciphers and stream ciphers, the primes p, q in RSA encryption and digital signature schemes, the nonce in most authentication protocols. In fact, producing a true random sequence is not possible or at least it is very costly. Therefore a *pseudorandom sequence* is used in cryptography instead.

A pseudorandom sequence is not a true random sequence, but it should be indistinguishable from a true random sequence. There are some properties that a pseudorandom sequence should fulfill to assure its randomness. Golomb in [25] proposed three randomness postulates to measure randomness of binary periodic sequences. For a sequence $S = s_0, s_1, ..., s_{N-1}$ with period N, the Golomb postulates are:

- **R-1:** In every period, the difference between the number of zeros and the number of ones should not exceed 1.
- **R-2:** In every period, half the runs (consecutive 0's or 1's) have length one, one-fourth have length two, one-eighth have length three, etc., as long as the number of runs so indicted exceeds 1. Moreover, for each of these lengths, there are equally many runs of 0's and 1's.
- **R-3:** For the *auto-correlation* function $C(\tau)$ we have:

$$C(\tau) = \begin{cases} N & \text{if } \tau \equiv 0 \ \ (\text{mod N}) \\ K & \text{if } \tau \not\equiv 0 \ \ (\text{mod N}), \end{cases} \tag{1}$$

  where K is a constant and the auto-correlation function, $C(\tau)$, for sequence $S = s_0, s_1, ..., s_{N-1}$ is defined by:

$$C(\tau) = \sum_{i=0}^{N-1} (-1)^{s_i + s_{i+\tau}}. \tag{2}$$

These properties are necessary for each pseudorandom sequence but they are not sufficient. Therefore more properties are defined in the literature. Randomness is a probabilistic property; that is, the properties of a random sequence can be characterized and described in terms of probability. To determine that a pseudorandom bit sequence fulfills these properties, we use *statistical tests*. Each statistical test measures certain aspects of the quality of the sequence.

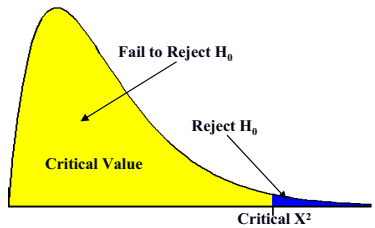

Fig. 1: *Critical Value for χ^2 Distribution.*

There are many statistical tests which can be applied to a sequence to compare and evaluate the randomness of the sequence. Because there are so many tests for judging whether a sequence is random or not, no specific finite set of tests is deemed "complete". In addition, the results of statistical testing must be interpreted with some care and caution to avoid incorrect conclusions about a specific generator. Generally, if the output of a sequence generator does not pass a statistical test, we can conclude that there is a distinguisher for this generator which can lead to a key recovery attack in some cases.

A statistical test is formulated to test a specific *null hypothesis* (H_0) which is "the sequence being tested is random" in cryptography. For each applied test, a decision or conclusion is derived from acceptance or rejection of the null hypothesis, i.e., whether the generator is (or is not) producing random values, based on the produced sequence.

To design a new statistical test, a relevant randomness *statistic* must be chosen and used to determine the acceptance or rejection of the null hypothesis. Under an assumption of randomness, such a statistic has a distribution of possible values which is determined by mathematical methods under the null hypothesis. To apply a test, a statistic value is computed on the tested data and compared to the *critical value*. If the test statistic value exceeds the critical value, the null hypothesis for

Table 1: *The status of statistical results*

TRUE SITUATION	**Conclusion**	
	Accept H_0	Accept H_a (reject H_0)
Data is random (H_0 is true)	No error	Type I error
Data is not random (H_a is true)	Type II error	No error

randomness is rejected. Otherwise, the null hypothesis (the randomness hypothesis) is not rejected. Figure 1 illustrates the critical value and rejection area for χ^2 Distribution.

When the calculated test statistic value exceeds the critical value, it means that the low probability event does in fact occur. But from a statistical hypothesis testing point of view, the low probability event should not occur naturally. Therefore, we can conclude that the original assumption of randomness is suspect or faulty and the statistical hypothesis testing yields; H_0 (randomness) is rejected and H_a (non-randomness) is accepted. Table 1 relates the true (unknown) status of the data at hand to the conclusion arrived at using the testing procedure.

In statistical tests, *Type I Error* occurs when the null hypothesis is rejected while the data is infact random. Additionally, *Type II Error* occurs when the null hypothesis is accepted while the data is non-random.

The probability of a type I error is often called *the level of significance* of the test and is denoted by α. So, α is the probability of a wrong result in the test for a sequence while it really is random. That is, a sequence appears to have non-random properties even when a "good" generator produced the sequence.

The probability of a type II error is denoted as β which means a "bad" generator produced a sequence that appears to have random properties. Unlike α, β is not a fixed value. β can take on many different values because there are an infinite number of ways that a data stream can be non-random, and each different way yields a different β. The calculation of the type II error, β, is more difficult than the calculation of α because of the many possible types of non-randomness. For designing a new statistical test, one should prove that the new test is consistent when the length of tested data goes to infinity. For this, it should be shown that β tends to zero when the length of tested data goes to infinity.

In paper I, we proposed a new statistical test which is suitable to apply to the output of all cryptographic primitives (e.g. hash function,

block cipher, stream cipher, random number generator, ...), and we proved that β goes to zero (i.e. the power of our test tends to one) when the length of tested data goes to infinity.

2.2 Stream Cipher

The *one-time pad* (or Vernam) cipher is the only perfect secure system which is unbreakable [56]. In this system, a new random symbol is produced independently from plaintext and previous symbols to encrypt each symbol of the plaintext. Therefore, the length of the secret key should be as long as the plaintext and it is hard to store and distribute this key between the trusted users. Therefore, it is an impractical system. In practice, we use an algorithm to produce a pseudorandom sequence from a short secret key but as long as needed. The short secret key can be produced and distributed easily by asymmetric key cryptography and then the trusted users can produce the same pseudorandom sequence to encrypt and decrypt messages.

Definition 1. *A* pseudorandom bit generator *(PRBG) is a deterministic algorithm which, given a truly random binary sequence of length k, outputs a binary sequence of length $l > k$ which "appears" to be random. The input to the PRBG is called the seed, while the output of the PRBG is called a pseudorandom bit sequence [48].*

One approach to designing a pseudorandom bit generator is stream ciphers. In cryptography, a stream cipher is a symmetric key cipher which produces a sequence of pseudorandom bits depending on a secret key (and usually an Initial Vector). This is combined with the plaintext bit stream typically by an exclusive-or (XOR) operation. Figure 2 shows the structure of a stream cipher. In contrast, a block cipher operates on each block of plaintext and modifies them depending on a secret key (and usually an Initial Vector in CFB or OFB modes).

Stream ciphers are widely used due to their speed and simplicity in hardware implementation and also they need lower power than other primitives. The foremost and simplest element used to design a stream cipher is a *Linear Feedback Shift Register* (LFSR) [25] which is usually combined with a nonlinear element. LFSRs can generate sequences with highly attractive properties in cryptography and can be implemented efficiently in hardware and software. An LFSR contains n cells which are shifted one cell in each step. The last cell is considered as an output bit and the first cell is modified depending on the value of others.

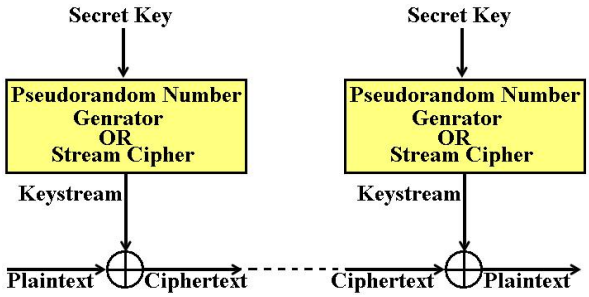

Fig. 2: *Structure of Stream Cipher.*

In other words, if we denote the state of the register at time $t-1$ by $S_{t-1} = (s_{t-1}, s_t, ..., s_{t+n-2})$, the next state S_t is determined by

$$S_t = (s_t, s_{t+1}, ..., \sum_{i=0}^{n-1} c_i s_{t+i-1}). \tag{3}$$

The weakness of an LFSR is its linearity. This can be fixed by including a nonlinear element. To use LFSRs in stream ciphers, there are several well known approaches to include nonlinearity:

- **Nonlinear Filter Generators:** In this structure, the state of the LFSR is passed through a nonlinear function to produce the output sequence. We refer to [4, 48] for further details. The *knapsack generator* is an example of a nonlinear filter generator.
- **Nonlinear Combination Generators:** Another approach to include a nonlinear element is the usage of a nonlinear function to apply on the output of several parallel LFSRs. The *Geffe generator* [28] and *Summation generator* [53] are two examples of this structure.
- **Nonlinear Feedback Shift Registers:** An other approach to designing a nonlinear stream cipher is the use of a nonlinear boolean function as a feedback for a shift register. In other words, instead of using a linear combination in (3), to modify the first cell, a nonlinear combination is used. Grain is an example using nonlinear feedback shift registers [35].
- **Clock-Controlled Generators:** Using irregular clocking instead of regular clocking in LFSR is another approach to include nonlinearity. There are several different substructures for clock-controlled

generators. We refer to [27] for further details. The *Alternating Step Generator* [26] and *Shrinking Generator* [19, 49] are two examples of this structure.

In papers II-IV, we present our results on attacks on several families of the Alternating Step Generator. These are modified versions to resist the previous attacks applied on the original alternating step generator.

2.3 Stream Cipher Cryptanalysis

Before using a stream cipher, its security should be proved and this is often done by proving its immunity against all known attacks. There are many attacks that apply to stream ciphers but a survey of two attacks which are relevant to this thesis are given in this section.

2.3.1 Algebraic Attack on Stream Ciphers

One powerful method of cryptanalysis is the *algebraic attack* [18, 45] which has received a great deal of attention in the literature during the last decade. This attack turns the problem of breaking the cipher into a problem of solving an equation system and if it is sufficiently difficult to solve, we can consider that the cipher is secure against this attack.

In this technique, each output symbol, z_t, for each clock t is represented as a polynomial in the initial state (or internal state) at the same time. Then an equation system is constructed with the polynomials and solved to find the initial state (or internal state).

Of course, one can also try to use the knowledge of the internal state to clock the stream cipher backwards and recover the secret key. Depending on the type of equation system, there are several methods to solve it, like Gröbner Bases, Linearization (system needs to be grossly overdefined), XL, XLS etc. Also, we can reduce the degree of equations according to our knowledge about the algorithm and reduce the complexity of the attack.

There are many types of algebraic attacks on stream ciphers, block ciphers and public key cryptosystems. We are interested in a version of especially an algebraic attack that is applicable to stream ciphers and to clock-controlled LFSR-based stream ciphers. In [3] an algebraic attack approach to a family of clock-controlled LFSR-based systems is presented. In this thesis, a similar approach combined with some new ideas are used to cryptanalyze some clock-controlled generators related

to the alternative step generator family. More details are presented in papers II-IV.

2.3.2 Differential Attack

Another general method for analyzing cryptographic primitives is *differential cryptanalysis* introduced in [9, 10]. It is usually a kind of *chosen plaintext attack*, meaning that encrypted ciphertexts can be obtained for some set of chosen plaintexts. The technique was widely applied to stream ciphers [6], block ciphers [8, 10], hash functions [52] and also public key cryptosystems [20, 24], and most of the new constructions are specifically designed to resist this attack.

The basic idea uses a pair of plaintexts which are related by a constant *difference*, (Δx). The exclusive-or (XOR) operation is usually considered to find the difference, but it can be defined in several ways depending on the algorithm. The attacker then calculates the difference of corresponding ciphertexts, (Δy). The resulting pair of differences (Δx, Δy) is called a *differential*. We are not interested in what exactly happens in the cipher when the desired differential occurs, but only interested in the probability of the differential. If the differences of the corresponding ciphertexts has any statistical patterns in their distribution, the cipher can be distinguished from random and it can lead to a key recovery attack. A differential of a stream cipher is a prediction that a given input difference (Key, IV or the internal state) produces some specific output differences (Key stream or internal state).

The differential attack can be used with *fault analysis* [7] which has several modes. In fault attacks, introduced by Boneh et al. in 1997 [11], the attacker can introduce errors during the computation, leading to an error in the output. The errors can be injected in several ways(e.g. X-ray, laser light, ion beams, ...) and in many places according to the algorithm. By examining the difference between an unfaulty computation and a faulty one, the attacker can deduce information on the computation. Excellent surveys on fault attacks can be found in [5, 29]. Differential fault attacks were used in various cryptanalytic attacks on stream ciphers [7] which is relevant to our work.

2.4 Secure Network Coding

Network coding was introduced by Ahlswede *et al.* in [12]. In this communication paradigm, network nodes are allowed not only to forward unmodified packets, as routers in a classical store-and-forward network are restricted to, but also to modify packets by performing mathematical operations to form new packets prior to forwarding them.

In [17], Cai and Yeung presented a certain security problem that can be alleviated by network coding. They introduced a model for secure linear network coding based on using secret sharing idea that achieved perfect information security against a wiretapper with access to a limited number of network links.

In [23], Feldman *et al.* showed that finding a matrix for the construction of an optimal secure network code is equivalent to finding a linear code with certain generalized distance properties and improved the lower bound of the field size. In this model, increasing the level of security results in the growth of the required field size which leads to an inefficient model in practice.

In [47], Lima *et al.* considered a different approach to provide secure network coding. They showed that linear network coding is sufficient to set up a secure network coding scheme when a network is constructed by imposing a limitation on the input degree of the nodes.

In this thesis, a new model, named *Two Layer Secure Network Coding* or 2-LSNC, for secure network coding is proposed. By this model, we improve on the number of links that a wiretapper needs to access in order to extract the secret message, the level of security and the cost for increasing the level of security. Our model is *scalable*, which means that the efficiency improves as network size grows while our simulations shows that this property does not hold for previous approaches. In other words, resistance against a more powerful wiretapper in [17] and [47] can be achieved by using a larger field size. Our method has no constraint on the field size and it is only necessary to use a field that gives a feasible network coding solution.

2.5 Secret Sharing

In cryptography, *secret sharing* refers to any method for distributing a secret (with a *dealer*) amongst a group of n participants (*players*), each of which is allocated a share of the secret. Secret sharing was invented independently in 1979 by A. Shamir [55] and G. Blakley [13]. The secret can be reconstructed only when a sufficient number of shares are combined together. In basic concept, (n,t)-scheme, the secret S is divided into n shares $(s_1, \ldots, s_n)$ in such a way that:

1. Knowledge of any t or more s_i makes S *easily* computable.
2. Knowledge of any $t-1$ or fewer s_i leaves S *completely undetermined*.

For example, in the *trivial* (n, n)-threshold scheme (i.e. t is equal to n), $n-1$ random numbers $(r_1, \ldots, r_{n-1})$ are generated as $n-1$ shares and for the last share we have: $r_n = S \oplus r_1 \oplus r_2 \oplus \ldots \oplus r_{n-1}$, where $\oplus$ is any discrete formal addition. It is straightforward to see that S could be reconstructed with knowledge of all shares, while no subset of $n-1$ or fewer shares can reconstruct the secret S.

In [55] a simple (n, t)-threshold scheme based on polynomial interpolation is proposed where the polynomials can be replaced by any other collection of functions which are easy to evaluate and to interpolate.

This technique is an important method used to distribute a secret key in symmetric cryptography. However, there are many different methods for secret sharing in the literature, and a new method suitable for secret network coding is presented in this thesis.

3 Summary of Papers

This thesis consists of eight papers. In the following sections, a short overview of each paper is given.

3.1 Paper I: New Statistical Box-Test and Its Power

In the first paper, we propose a new statistical box-test whose main idea is to compare the distribution of repeated patterns in a given sequence which can be the output of any cryptographic primitive. By a hypothesis test, this distribution in the tested data is compared with its expected value in a true random sequence.

The basic idea comes from the *Empty Box Test* which was proposed by David in [21]. Let μ_0 denotes the number of patterns that never appear in the tested sequence. In the case of a true random sequence, the variable μ_0 has a normal distribution and the empty box test evaluates the normality of μ_0. To the best of our knowledge it is the first time that this notion is used in cryptography. We also extended this idea to drive several tests based on μ_r for $r \geq 0$. Additionally, we proposed a new statistical test based on a nonlinear combination of μ_r for $r \geq 0$.

3.2 Paper II-IV: Algebraic Attacks on some modified versions of the Alternating Step Generator

From a cryptanalysis point of view, a good stream cipher should be resistant against a *known-plaintext attack*. In this kind of attack, the cryptanalyst is given a plaintext and the corresponding ciphertext, and the task is to determine the secret key. For a synchronous stream cipher, this is equivalent to the problem of finding the secret key or initial state that produces a given keystream output.

In papers II-IV, some modified versions of the *Alternating Step Generator* (ASG) are discussed and analyzed. An alternating step generator, a well-known stream cipher proposed in [26], consists of three linear feedback shift registers, **A**, **B** and **C**, whose lengths are l, m and n respectively. The first LFSR, **A**, is clocked regularly and the two others, **B** and **C**, are clocked in a Stop/Go manner [15, 27]. At each time, the clock-control bit from **A** determines which one of the two Stop/Go LFSRs is clocked, and the output sequence is obtained as a bit-wise sum of the two Stop/Go clocked LFSR sequences. Figure 3 shows the simple structure of the original Alternating Step Generator.

In all these papers, we used a modified version of algebraic attack combined with a new idea in each paper to reduce the complexity of our attack on a specific class of the ASG family. In all papers, we search over all possible initial states of the regularly clocked binary LFSR, **A**, and calculate some equations based on unknown initial states of the irregularly clocked binary LFSRs, **B** and **C**, and known output keystream sequence to construct an equation system. In the original concept of the algebraic attack, we have to solve this equation system, but we guess a bit of LFSR **B** and find an irregular output of each LFSR, **B** and **C**, with a quite low complexity. By using the irregular outputs found in the first part of the attack, we find the initial states

of the LFSRs, **B** and **C**, which is done in different ways in each paper presented briefly in the following subsections.

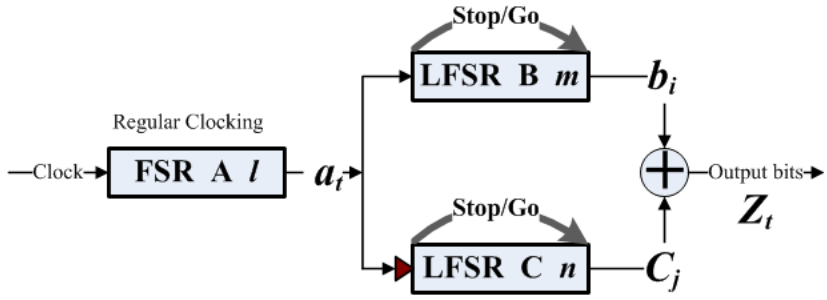

Fig. 3: *Structure of the original Alternating Step Generator (ASG).*

3.2.1 Paper II: Algebraic Attack on the Alternating Step(r,s) Generator

The Alternating Step(r,s) Generator [40], ASG(r,s), is a clock-controlled sequence generator proposed in 2002. This is similar to the original alternating step generator with one exception; the registers **B** (resp. **C**) are clocked r times (or not clocked) (resp. s times or not clocked) depending on the clock-control bit from the first register. The designer claimed there is no efficient attack against the ASG(r,s) since r and s are kept secret. Figure 4 illustrates the structure of ASG(r,s).

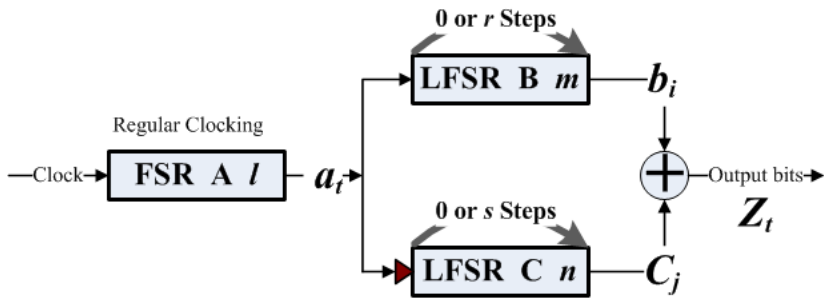

Fig. 4: *Structure of the Alternating Step(r,s) Generator (ASG(r,s)).*

Our contribution in this paper is a new model for ASG(r,s) which is independent of the secret parameters r and s which can reproduce

the same output but with a different initial state. We apply our efficient algebraic attack on this model instead of the original algorithm. Using our result from applying our attack on this model, we can find the secret key as well as parameters r and s. In total, our attack needs $3(m+n)$ bits of the output sequence to find the secret key with $O((m^2+n^2)2^{l+1} + m^3 2^{m-1} + n^3 2^{n-1})$ computational complexity. We show that this system is no more secure than the original ASG, in contrast to what was claimed by the ASG(r,s)'s constructor.

3.2.2 Paper III: Algebraic Attack on the More Generalized ASG and Modified ASG

The More Generalized Clock-Controlled Alternating Step Generator [41], MGASG, is a clock-controlled sequence generator proposed in 2004. It is similar to the original alternating step generator with one exception; at each time unit t, the registers **B** (resp. **C**) clocked $r(t)$ times or not clocked (resp. $s(t)$ times or not clocked) depending on the clock-control bits in the first register. At each time t, the values of $r(t)$ and $s(t)$ are determined according to the values of W_B and W_C bits of the first register respectively. The special case when $(r(t) = r)$ and $(s(t) = s)$ is the Alternating Step(r,s) Generator discussed in paper II, and the special case when $r(t) = s(t) = 1$ is the original Alternating Step Generator. Again the designer claimed that there is no efficient attack against the MGASG since the positions of controller bits and the values of the parameters W_B and W_C are kept secret and therefore, $r(t)$ and $s(t)$ are unknown. The structure of this algorithm is illustrated in Figure 5.

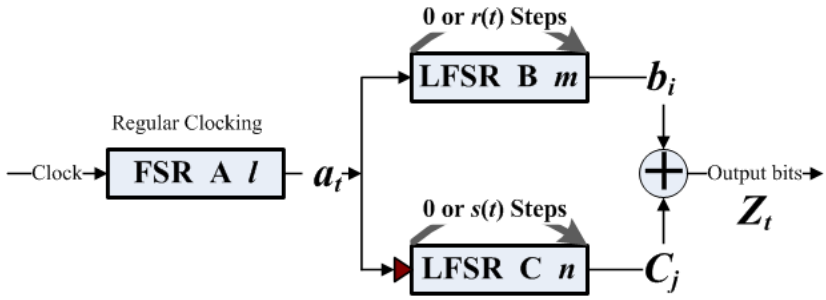

Fig. 5: *Structure of the More Generalized Clock-Controlled Alternating Step Generator.*

The Modified Clock-Controlled Alternating Step Generator, MASG, proposed in 2009, is similar to the More Generalized Clock-Controlled Alternating Step Generator [41], MGASG, with one exception; the output keystream is produced by a bit wise sum of the output keystream of all three registers instead of only two irregular registers. Figure 6 illustrates the structure of this algorithm.

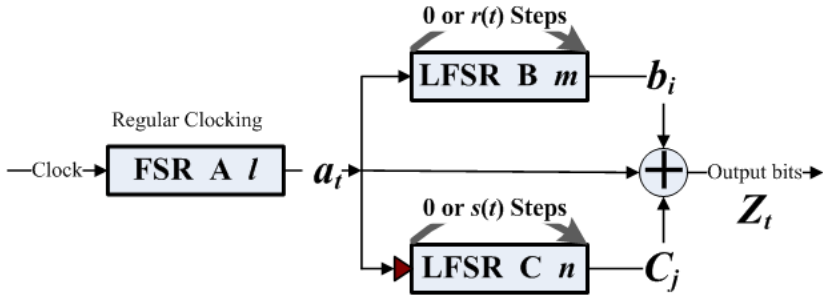

Fig. 6: *Structure of the Modified Clock-Controlled Alternating Step Generator.*

Our contribution in this paper is the combination of *MAP Decoding on the Deletion Channel* presented by Johansson in [39] and our modified algebraic attack to achieve an applicable attack on these structures. Our attack on these structures use a few bits of the output sequence to find the secret key with a computational complexity of $O(lM^2 2^{M+l+6}(W_B + W_C))$ while $M = \max(m, n)$. In the case of $m = n = l = 64$ and $W_B = W_C = 8$, our attack can find the secret key using 512 output bits and a complexity of $O(2^{156})$ steps, while the author claims that the best attack needs $O(2^{665.8})$ steps and the exhaustive search needs $O(2^{774.8})$ steps. The designer of MASG claims that the LCT attack needs $O(2^{669.5})$ steps for our example.

3.2.3 Paper IV: Algebraic Attack on the Second Class of Modified Alternating $\vec{k}$-Generators

Alternating $\vec{k}$-generators [16] is a family of binary clock-controlled keystream generators which fixes the weaknesses of the previous alternating step generator's families by inserting a nonlinear element in three ways and proposes three different classes. In this paper, we discuss the second class of modified alternating $\vec{k}$-generators. This generator is similar to the original ASG structure with one exception; the controller bit is produced by applying a nonlinear boolean function

on the internal state of the controller LFSR, **A**. This algorithm is illustrated in Figure 7. The designer proposed another stronger version of this structure by changing the output function. In the stronger version, instead of two LFSRs, the output of all three LFSRs are combined to produce the output keystream.

In this paper, we applied our algebraic attack combined with the Berlekamp-Massey algorithm on both versions of the modified alternating $\vec{k}$-generators. The computational complexity of our attack is $O(2^{l+1}(m^2+n^2))$ (resp. $O(2^{l+1})$) if the feedback polynomials of the generating registers are unknown (resp. known).

Johansson's attack is the best previous attack which can be applied on this structure and its complexity is less than the complexity of our attack, but it needs many more keystream bits. Additionally, if the feedback polynomial of the controlled registers are unknown, Johansson's attack (and also all previous attacks) is not applicable, but our attack can be applied. Our results show that the security of the second class of $\vec{k}$-generators is not better than the security of the original alternating step generator against our algebraic attack.

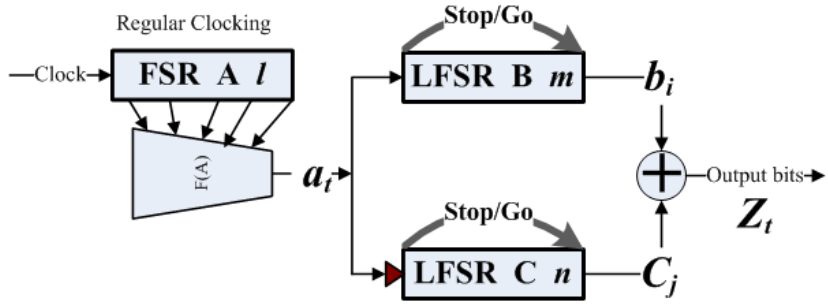

Fig. 7: *Structure of the second class of modified alternating $\vec{k}$-generators.*

3.3 Paper V: Security Analysis of the Step(D,K) Generator with Respect to its Parameters

Ciphers based on an irregularly clocked LFSR are one of the main and widely used types of stream ciphers. The simplest scheme uses only two LFSRs; the first one is clocked regularly and its output controls the clocking of the second one which produces irregular output sequence. In general, the second register is clocked D or K times which is called a

Step(D,K) Generator and illustrated in Figure 8. Most of the well known clock-controlled structures are special cases of the Step(D,K) Generator, e.g. Stop/Go generator is a Step(0,1) Generator.

In this paper, we discuss the security of the Step(D,K) Generator with respect to its parameters D and K. We will calculate the probability, P(n), that the n^{th} bit of the regular sequence, generated by the second LFSR, appears in the irregular output sequence. We will show that if P(n) is zero for some values of n, we can reduce the time complexity of the general attacks. In the case of the correlation attack based on the *Levenshtein Distance*[30], we will show how we can improve the time complexity of the attack. Finally, some recommendations will be presented to answer the question of choosing good parameters.

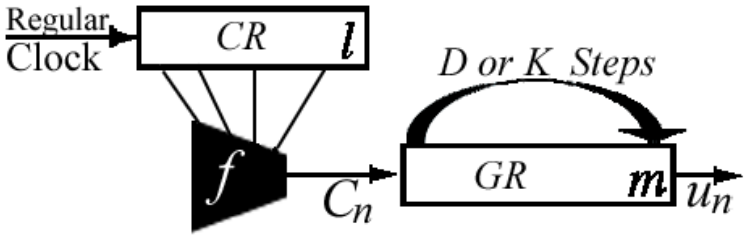

Fig. 8: *Structure of the Step(D,K) Generator.*

3.4 Paper VI: Differential Distinguishing Attack on the Shannon Stream Cipher Based on Fault Analysis

The Shannon stream cipher was proposed by Philip Hawkes *et al.* [33] as a software-efficient algorithm, with key length up to 256 bits. The Shannon stream cipher is an entirely new design, influenced by members of the SOBER family stream cipher [34]. It consists of a single nonlinear feedback shift register of length 16 with 32-bit words in each cell. There is also an extra word, which is supplemented for message authentication with 32 parallel CRC-16 registers. The keystream generator part of the Shannon stream cipher is illustrated in Figure 9.

The contribution of this paper is proposing a differential distinguisher for the Shannon stream cipher and finding a new weakness for the

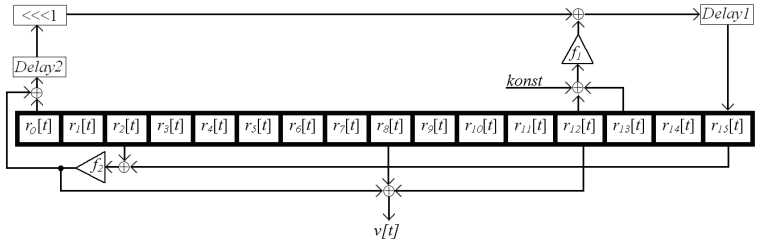

Fig. 9: *Structure of the Shannon Stream Cipher.*

nonlinear function used in the keystream generator. The distinguisher's complexity is four times the complexity of running the Shannon stream cipher and our attack only needs two differential outputs for a successful attack with error probability equal to 0.001. Another contribution of this paper is using the fault analysis method for distinguishing the output of the Shannon stream cipher.

3.5 Paper VII: Two Layer Secure Network Coding - (2-LSNC)

Two important parameters for network users are security and cost of network resource utilization. From the cost perspective, network coding promises a higher gain compared to ordinary routing. In this paper, a new model for secure network coding is proposed. This model is named *Two Layer Secure Network Coding* or 2-LSNC. The advantage of our new model is three-fold: 1) The number of links required for the wiretapper to extract the secret is increased. 2) For scalability analysis we have established a metric called *level of security*, and we demonstrate that our model scales well with respect to this metric. 3) The tradeoff between the cost and the level of security is analyzed. In our model, this cost is lower than in Cai and Yeung's model [17] when the network size and the number of sinks reaches a critical point. Our approach is compared with other models by simulation.

3.6 Paper VIII: Wiretapping Based on Node Corruption over Secure Network Coding: Analysis and Optimization

A new type of attack on secure network coding is introduced in this paper. In this model, network nodes, which handle the traffic from the source node to sink nodes are potentially viewed to be corruptible. We study the maximum security capacity for this problem for a single-source single-sink scenario, and we generalize our study for multicast with network coding. Based on our study, two optimization problems are introduced to increase the security against the attacks under study. We have shown by simulation results that our proposed optimization method has increased the security against node corruption considerably, and at the same time, the cost per level of security is lower compared to optimization methods without constraints on node corruption.

4 Future Research

- In **Paper I**, a new statistical test is presented. In this paper, we showed that about one-third of the output blocks occur more than one time and the same proportion of the output patterns do not occur at all in the output. In the case of hash functions, this idea may help to reduce the complexity of finding a collision or second pre-image. Therefore, this can be a future topic for research.

- In **Paper II-IV**, an algebraic attack on the several families of Alternating Step Generator are presented. Our attack was applicable due to the linearity of the output function in all analyzed algorithms. Finding an attack when a nonlinear function is used to produce the output sequence can be another topic for future research.

- In **Paper V**, we discuss the security of the step(D,K) generator with respect to its parameters. Recently, *Jump Register* [37, 38] is used in some famous stream ciphers, e.g. Pomaranch [36] and Mickey [14] stream ciphers. The last topic that we propose for future work is investigation of the applicability of this idea on jump registers and then on the Pomaranch and Mickey stream ciphers.

REFERENCES

[1] Al-Kindi, "Cryptgraphy, Codebreaking and Ciphers", Retrieved 12 January 2007.

[2] Ibrahim A. Al-Kadi, "The origins of cryptology: The Arab contributions", *Cryptologia* 16(2): 97-126, April 1992.

[3] S. Al-Hinai, L. Batten, B. Colbert, and K. Wong, "Algebraic Attacks on Clock-Controlled Stream Ciphers", *Lecture Notes in Computer Science* (LNCS), Volume 4058, pages 1-16, Springer Berlin, Heidelberg, 2006.

[4] G. Ars, "Une application des bases de gröbner en cryptographie", DEA de Rennes I, 2001.

[5] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerer's apprentice guide to fault attacks", Proceedings the IEEE 94(2):370-382, 2006. Earlier version in Proc. of FDTC 2004.

[6] Eli Biham, O. Dunkelman, "Differential Cryptanalysis in Stream Ciphers", Technical report CS-2007-10. `http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi?2007/cs/cs-2007-10`

[7] Eli Biham and Adi Shamir, "Differential Fault Analysis of Secret Key Cryptosystems", Advances in Cryptology - CRYPTO '97, LNCS vol. 1294, pp. 513-525, Springer-Verlag, 1997.

[8] Eli Biham and Adi Shamir, "Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer", Advances in Cryptology – CRYPTO '91. Springer-Verlag. pp. 156-171, 1991.

[9] Eli Biham and Adi Shamir, "Differential Cryptanalysis of the Data Encryption Standard", Springer-Verlag, 1993.

[10] Eli Biham and Adi Shamir, "Differential Cryptanalysis of DES-like Cryptosystems". Advances in Cryptology – CRYPTO '90. Springer-Verlag. pp. 2-21, 1990.

[11] D. Boneh, R.A. DeMillo, and R.J. Lipton. "On the importance of eliminating errors in cryptographic computations", *Journal of Cryptology* 14(2):101-119, 2001. Extended abstract in Proc. of EUROCRYPT'97.

[12] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", *IEEE Transactions on Information Theory*, Vol. 46,

April 2000, pp. 1204-1216.

[13] G. R. Blakley, "Safeguarding cryptographic keys", *Proc. the National Computer Conference,* Vol. 48, 1979, pp. 313-317.

[14] Steve Babbage and Matthew Dodd, "The MICKEY Stream Ciphers", LNCS, vol. 4986, pages 191-209, Springer, and the ECRYPT/eSTREAM project: pp. 224-243, 2008.

[15] T. Beth and F. Piper, "The Stop and Go Generator", Advances in Cryptology: Proceedings of Eurocrypt'84, LNCS, Berlin: Springer-Verlag , vol. 209, pp. 88-92, 1985.

[16] R. Bialota, G. Kawa, "Modified Alternating $\vec{k}$-Generators", Designs, Codes and Cryptography, Volume 35, Number 2, May 2005 , pp. 159-174(16), Springer Science+Business Media, Inc. Manufactured in The Netherlands.

[17] N. Cai, and R. W. Yeung, "Secure network coding", *Proceedings of IEEE Symposium on Information Theory*, 2002.

[18] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, "Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations", Advances in Cryptology - Eurocrypt 2000 1807 (2000), pp. 392-407.

[19] D. Coppersmith, H. Krawczyk, and Y. Mansour, "The Shrinking Generator", CRYPTO'93, *Lecture Notes in Computer Sciences*, vol. 773, pp. 22-39, Springer, Berlin, 1993.

[20] Vivien Dubois, Pierre-Alain Fouque, Jacques Stern, "Cryptanalysis of SFLASH with Slightly Modified Parameters", Advances in Cryptology, proceedings of *EUROCRYPT* 2007, *Lecture Notes in Computer Science* 4515, pp. 327-341, Springer, 2007.

[21] F.N. David, *Two combinatorial tests whether a sample has come from a given population,* Biometrika, vol. 37(1950), 97–110.

[22] eSTREAM, the ECRYPT Stream Cipher Project, 2004-2008. `http://www.ecrypt.eu.org/stream/`

[23] J. Feldman, T. Malkin, R. A. Servedio, and C. Stein, "On the Capacity of Secure Network Coding", *Proc. 42nd Annual Allerton Conference on Comunication, Control and Computing,* September 2004.

[24] Pierre-Alain Fouque, Louis Granboulan, Jacques Stern, "Differential Cryptanalysis for Multivariate Schemes", Advances in Cryptology, proceedings of *EUROCRYPT* 2005, *Lecture Notes in Computer Science* 3494, pp. 341-353, Springer, 2005.

[25] S.W. Golomb, "Shift Register Sequences", Revised Edition, Aegean Park Press, 1982, Chapter 3.

[26] C. G. Günther, "Alternating Step Generators Controlled by De Bruijn Sequences". Advances in Cryptology: Eurocrypt 87, LNCS, Spingler-Verlag, vol. 309, pp. 5-14, 1988.

[27] D. Gollmann and W. Chambers, "Clock-Controlled Shift Register: A Review", *IEEE J. Selected Areas Communications*, vol. 7, NO. 4, pp. 525-533, May 1989.

[28] P. R. Geffe, "How to protect data with ciphers that are really hard to break", Electronics, pp. 99-101, Jan. 1973.

[29] C. Giraud and H. Thiebeauld, "A survey on fault attacks", In J.-J. Quisquater et al., editors, Smart Card Research and Advanced Applications VI (CARDIS 2004), pages 159-176. Kluwer Academic Publishers, 2004.

[30] J. Golic, and M. Mihaljevic, "A Generalized Correlation Attack on a Class of Stream Ciphers Based on the Levenstein Distance", *Journal of Cryptology*, 3, 1991, pp. 201-212.

[31] Mehdi M. Hassanzadeh, Elham Shakour and Shahram Khazaei, "Improved Cryptanalysis of Polar Bear", State of Art of Stream Ciphers (SASC'06), pages 154-160, Feb. 2006, Leuven, Belgium. Submitted at 2005-12-24 in ECRYPT website. `http://www.ecrypt.eu.org/stream/papersdir/084.pdf`

[32] Mehdi M. Hassanzadeh, Shahram Khazaei,"On IV setup of Pomaranch", State of Art of Stream Ciphers (SASC'06), pages 7-12, Feb. 2006, Leuven, Belgium. Submitted at 2005-12-11 in ECRYPT website. `http://www.ecrypt.eu.org/stream/papersdir/082.pdf`

[33] P. Hawkes, C. McDonald, M. Paddon, G. Rose, M. Wiggers de Vries, "Design and Primitive Specification for Shannon", Dagstuhl Seminar Proceedings 07021, Symmetric Cryptography. http://drops.dagstuhl.de/opus/volltexte/2007/1019

[34] P. Hawkes, G. Rose, "The t-class of SOBER stream ciphers", Technical Report, QUALCOMM Australia,1999. www.qualcomm.com.au

[35] M. Hell, T. Johansson and W. Meier, "Grain - A Stream Cipher for Constrained Environments", ECRYPT Stream Cipher Project Report 2005/010, 2005, available at http://www.ecrypt.eu.org/stream/

[36] Cees J.A. Jansen, Tor Helleseth, Alexander Kholosha, "Cascade Jump Controlled Sequence Generator and Pomaranch Stream Cipher", LNCS, volume 4986, pages 224-243, Springer, 2008.

[37] C.J.A. Jansen, Partitions of polynomials: "Stream ciphers based on jumping shift registers", Cardinal, J., Cerf, N., Delgrange, O., Markowitch, O. (eds.) 26th Symposium on Inf. Theory in the Benelux, Enschede, Werkgemeenschap voor Informatie- en Communicatietheorie, pp. 277-284, 2005.

[38] C.J.A. Jansen, "Stream cipher design based on jumping finite state machines", Cryptology ePrint Archive, Report 2005/267 (2005), http://eprint.iacr.org/2005/267/.

[39] T. Johansson, "Reduced Complexity Correlation Attacks on Two Clock-Controlled Generators", In ASIACRYPT'98, pp. 342-356, 1998.

[40] A. Kanso, "The Alternating Step(r,s) Generator", SECI02, Tunis, Sep. 2002.

[41] A. Kanso, "More Generalized Clock-Controlled Alternating Step Generator", in: Proceedings of ACNS'04, Yellow Mountain, China, 8-11 June, LNCS 3089, Springer, Berlin, pp. 326-338, 2004.

[42] Shahram Khazaei, Mahdi M. Hassanzadeh and Mohammad Kiaei, "Distinguishing Attack on Grain", submitted 2005-10-14. http://www.ecrypt.eu.org/stream/papersdir/071.pdf

[43] Shahram Khazaei and Mahdi M. Hassanzadeh, "Linear Sequential Circuit Approximation of the TRIVIUM Stream Cipher", submitted 2005-09-27. http://www.ecrypt.eu.org/stream/papersdir/063.pdf

[44] Shahram Khazaei and Mehdi M. Hasanzadeh and Mohammad S. Kiaei, "Linear Sequential Circuit Approximation of Grain and Trivium Stream Ciphers", Cryptology ePrint Archive, submitted 2006-04-11, http://eprint.iacr.org/2006/141

[45] A. Kipnis and A. Shamir, "Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization", Advances in Cryptology - Crypto 1999 1666 (1999), 19-30.

[46] Valentin F. Kolchin, Boris A. Sevast'yanov and Vladimir P. Chistyankov, "Random Allocations",V. H. Winston & sons, Washington, D.C., 1978.

[47] L. Lima, M. Medard, and J. Barros, "Random linear network coding: a free cipher?", *Proceedings of IEEE Symposium on Information Theory*, 2007.

[48] P. van Oorschot A. Menezes and S. Vantome, "Handbook of Applied Cryptography", `http://www.cacr.math.uwaterloo.ca/hac/,1996`.

[49] W. Meier and O. Staffelbach, "The self-shrinking generator", In A. De Santis, editor, Advances in Cryptology - Eurocrypt'94, *Lecture Notes in Computer Sciences*, vol. 950, pp. 205-214, Springer, Berlin, 1995.

[50] National Institute of Standards and Technology of the United States; `http://www.nist.gov`

[51] NESSIE (New European Schemes for Signatures, Integrity and Encryption), 2000-2003. `https://www.cosic.esat.kuleuven.be/nessie/`

[52] Bart Preneel , Rene Govaerts , Joos Vandewalle, "Differential cryptanalysis of hash functions based on block ciphers", Proceedings of the 1st ACM conference on Computer and communications security, pp.183-188, November 03-05, 1993, Fairfax, Virginia, United States.

[53] Rainer A Rueppel, "Correlation immunity and the summation generator", *Lecture Notes in Computer Sciences*; 218 on Advances in cryptology-CRYPTO'85, pp.260-272, June 1986, Santa Barbara, California, United States.

[54] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22, May 15, 2001.

[55] A. Shamir, "How to share a secret", *Communications of the ACM,* Vol. 22(1), 1979, pp. 612-613.

[56] Claude E. Shannon, "Communication theory of secrecy system", Bell Syst. Tech. J. vol.28-4, pp. 656-715, 1949.

[57] Simon Singh, "The Code Book", pp. 14-20.